



TREBALL FINAL DE GRAU



ESCOLA
POLITÈCNICA SUPERIOR
UNIVERSITAT DE LLEIDA
INSPIRING THE FUTURE

Estudiant: Alexandru Martinas

Titulació: Grau en Enginyeria Informàtica

Títol de Treball Final de Grau: **Implementació d'un sistema de validació i anàlisis de funcions i scripts d'ús acadèmic de Matlab**

Director/a: **Josep Lluís Lèrida Monsó**

Presentació

Mes: Juliol

Any: 2018

Agraïments

M'agradaria dedicar unes paraules d'agraïment a totes les persones que han fet que aquest TFG sigui possible.

Des de petit, a mesura que anava avançant en la meva formació, me'n vaig adonar de que a mi m'agradava molt el món de la informàtica. Gràcies a un professor anomenat Ramón Sardà vaig poder decidir-me i cursar el grau d'Enginyeria informàtica.

També m'agradaria agrair als meus pares per tot el suport que m'han proporcionat al llarg d'aquests anys, per saber aconsellar-me i per animar-me en els moments més difícils que he tingut durant la carrera.

Per finalitzar m'agradaria donar les gràcies al meu tutor Josep Lluís, qui m'ha anat ajudant a resoldre petits dubtes que tenia sobre el projecte i qui m'ha anat guiant durant tot el procés.

Índex general

Resum	6
1. Introducció.....	7
1.1. Objectius	8
1.2. Motivació	9
1.3. Temporalitat	10
2. Anàlisis i Disseny.....	12
2.1. Anàlisis de Requeriments.....	13
2.1.1. Requeriments funcionals.....	13
2.1.2. Requeriments no funcionals.....	14
2.2. Decisions de disseny	14
2.2.1. Plataforma	15
2.2.2. Algoritmes.....	17
2.3. Disseny de la base de dades	20
2.3.1. Taula de login.....	20
2.3.2. Taula de activity.....	21
2.4. Rols.....	22
2.5. Diagrama de classes.....	22
3. Disseny de la interfície.....	24
3.1. Interfícies del professor/administrador.....	24
3.2. Diferències en les interfícies de l'alumne.....	31
4. Implementació.....	31
4.1. Llenguatge emprat: Java.....	32
4.2. JSP	32
4.3. Apache TomCat Server	33
4.4. Entorn de desenvolupament	34
5. Conclusions.....	34
5.1. Conclusions finals.....	35

5.2.	Valoració i autocrítica	36
5.3.	Treballs futurs o possibles ampliacions	37
6.	Bibliografia	38

Índex de figures

Figure 1: Diagrama BD	20
Figure 2: Diagrama de classes de la plataforma.....	23
Figure 3: Interfície de login.....	24
Figure 4: Interfície principal del professor/administrador.....	25
Figure 5: Interfície de la llista d'activitats.....	25
Figure 6: Interfície de creació d'activitats.....	26
Figure 7: interfície de Single activity test	27
Figure 8: Interfície de Multiple activity test	27
Figure 9: Interfície de resultats.....	28

Índex de taules

Taula 1: Planificació de les tasques principals	10
Taula 2: Requeriments funcionals	13
Taula 3: Requeriments no funcionals.....	14

Resum

La plataforma implementada pretén ajudar tant els professors com els alumnes d'assignatures a on es treballi amb l'eina Matlab. Aquesta plataforma serà capaç de validar les solucions dels alumnes amb la màxima eficiència possible. L'alumne podrà pujar al sistema una solució a l'exercici i obtenir un informe amb el resultat d'aquest. L'eina executa el codi de l'alumne a través d'un script de prova definit pel professor i compara la solució obtinguda amb la solució que el professor ha definit com a correcta. El professor de l'assignatura, per la seva banda, podrà crear diferents activitats per a que els alumnes validin les seves solucions. El professor podrà també pujar un conjunt de solucions per a una activitat i obtenir un informe de resultats per a totes elles.

Aquesta eina facilita que l'alumne rebi un feedback immediat sobre la seva solució, de manera que pot corregir i revisar les seves propostes fins a arribar a una solució similar a la que planteja el professor. Aquest feedback immediat permet a l'alumne progressar en l'aprenentatge-a auto-avaluar-se, etc...

En quant al professor, permet valorar ràpidament gran nombre d'activitats, així com la participació de l'alumne.

En aquest document es detallarà el desenvolupament d'aquesta plataforma, s'explicaran les eines emprades a la hora de programar-la, les decisions de disseny que s'han pres, les conclusions, possibles ampliacions, seguint una sèrie de fases i anàlisis de requeriments.

1. Introducció

Avui en dia, els professors que tenen assignades assignatures relacionades amb Matlab necessiten una eina que sigui capaç de corregir els exercicis/pràctiques dels alumnes amb una major eficiència ja que es perd molt temps anant un per un. La idea de la plataforma que s'ha implementat per a aquest treball es que tant els professors com els alumnes siguin capaços de validar la feina feta per l'alumne.

El que s'ha volgut aconseguir és:

- Que pugui servir per a que els propis alumnes provin les seves solucions i retornar un feedback immediat. En aquest moment, quan l'alumne entén que el seu exercici està be, ho pot contrastar enviat la solució al servidor per tal de rebre l'informe i poder revisar si es correcte del tot. La idea principal és que aquest informe serveixi de feedback immediat per a l'alumne i ajudar-lo a corregir i millorar els seus exercicis, a ser més autocrític i ser capaç d'autoavaluar-se.
- Que el professor pugui crear una sèrie d'activitats durant el curs per a als alumnes i tenir una nota en funció de la qualitat de la solució d'aquests (ja que fins ara el professor havia de proposar les activitats com a exercicis voluntaris d'aprenentatge perquè no se'n podia fer una avaluació).
- Que permeti solucionar alguns dels inconvenients de les proves avaluable de laboratori que es fan normalment en les assignatures de Matlab (com per exemple la lentitud a l'hora de corregir aquestes proves). Que automatitzi la correcció validant si l'script fa el que ha de fer en quina mesura i descartar automàticament tots aquells que directament no es poden executar.
- Obtenir informació sobre els resultats dels scripts dels alumnes per poder fer-ne una valoració per part del professorat i poder fer un seguiment del progrés de l'alumne.

A continuació es detallaran els objectius, la motivació i la temporització que s'ha emprat per tal de desenvolupar la plataforma i el treball en si.

1.1. Objectius

El principal objectiu del projecte es aconseguir desenvolupar una plataforma capaç de simplificar el procés de validació i/o correcció de funcions/scripts/codi en Matlab. Els objectius generals serien els següents:

- Dissenyar una plataforma capaç de gestionar la validació de les entregues d'estudiants de Matlab.
- Fer que el sistema sigui eficaç i intuïtiu a l'hora de realitzar les seves tasques (explicades en l'apartat 4 d'aquest treball).
- Fer que l'alumne sigui capaç d'autoavaluar-se mitjançant la plataforma, rebent un feedback immediat per part d'aquesta un cop executada la validació.
- Facilitar la correcció d'exercicis avaluable als professors mitjançant validacions múltiples i els resultats en forma de documents.

També s'han contemplat una sèrie d'objectius secundaris i personals relacionats amb el desenvolupament d'aquesta plataforma web:

- Aprendre a utilitzar correctament la tecnologia JSP (Java Server Page), juntament amb el seu desplegament en servidors (com per exemple GlassFish Server, TomCat Server...).
- Millorar el meu coneixement sobre Java, sobre tot enfocat al desenvolupament d'aplicacions web.

1.2. Motivació

Al començament es va plantejar la idea de desenvolupar un sistema que sigui capaç de determinar si existia plagi en les entregues d'exercicis avaluables de Matlab. Per tal d'implementar aquesta funcionalitat es va determinar que seria necessari implementar una correcta i eficaç plataforma que fos capaç de validar les entregues dels alumnes, retornant un feedback immediat en forma de document.

A causa de la magnitud del projecte s'ha determinat que les següents funcionalitats (plantejades com a objectiu quan es va començar aquest treball) s'haurien d'abordar en un treball futur. Aquestes funcionalitats són les següents:

- Verificació del codi per determinar si segueix les normes d'estil.
- Detecció de plagi entre les entregues dels alumnes d'una mateixa pràctica/activitat.

Personalment, el que més em va motivar a l'hora de realitzar aquest treball era:

- El fet de saber que aquesta plataforma es podria arribar a implementar en casos reals
- Que pogués facilitar molt la correcció de molts exercicis avaluables als professors.
- Que ajudaria molt als alumnes per tal de poder autoavaluar-se els seus propis exercicis.

1.3. Temporalitat

S'ha determinat que una divisió per tasques era l'opció més correcta a l'hora d'estimar el temps necessari per desenvolupar la plataforma de forma satisfactòria. La principal divisió que s'ha emprat es basa en assignar una certa quantitat de dies a cada tasca.

En la següents taula es poden observar les tasques a realitzar i la seva planificació temporal:

Tasca	Planificació
Preparació del projecte	
Instal·lar recursos necessaris	22/01/18 - 28/01/18
Estudi i pràctica dels recursos relacionats amb el projecte	29/01/18 - 09/02/18
Planificació del projecte	08/02/18 - 09/02/18
Anàlisi del projecte	18/02/18 - 15/03/18
Desenvolupament del projecte	
Implementació de la plataforma	10/02/18 - 11/06/18
Tests de la plataforma	10/02/18 - 14/06/18
Desplegament al servidor	28/02/18 - 31-02/18
Tests finals	04/05/18 - 15/06/18
Presentació del projecte	
Documentar tot el desenvolupament	16/02/18 - 03/06/18
Redactar la memòria	14/02/18 - 06/06/18

Taula 1: Planificació de les tasques principals

La planificació està dividida en 3 apartats. Preparació del projecte, Desenvolupament del projecte i Presentació del projecte. En el primer apartat s'han contemplat totes les tasques necessàries per tal de poder començar a desenvolupar la plataforma i determinar com s'havia d'encarar correctament el projecte. En el segon apartat hi estan presents les tasques relacionades amb la implementació de la plataforma. El tercer apartat està totalment enfocat a la redacció de la memòria.

Una cosa que s'ha de tenir en compte es que la planificació que s'acaba de mostrar es variable, ja que durant el desenvolupament de la plataforma han pogut sorgir problemes que obligaven a haver de dedicar més temps a certes tasques.

2. Anàlisis i Disseny

Per tal de que la plataforma funcioni correctament i es pugui adaptar a les necessitats dels usuaris (en aquest cas dels professors i alumnes), s'ha de fer un anàlisis de requeriments. El llistat de tots els requeriments s'ha distribuït en la típica divisió de requeriments funcionals (principals requeriments que ha de disposar la plataforma) i no funcionals (fan referència als requeriments més generals d'una aplicació. Son més secundaris, però igual d'importants que els funcionals).

2.1. Anàlisi de Requeriments

2.1.1. Requeriments funcionals

Requeriment	Descripció del requeriment
RF1	La plataforma ha de ser capaç de llegir i validar les solucions dels alumnes en base a la solució del professor.
RF2	La plataforma ha de permetre que, tant els alumnes com els professors, puguin pujar els codis/solucions a l'aplicació.
RF3	La plataforma ha de permetre a l'alumne veure el resultat de la validació.
RF4	La plataforma ha de ser capaç de crear un petit informe per al professor, quan s'intenta validar més d'una solució a la vegada.
RF5	La plataforma ha de poder executar les solucions dels alumnes amb l'script proporcionat pel professor.
RF6	La plataforma ha de ser capaç de generar documents PDF amb els resultats de la validació.
RF7	La plataforma ha de ser capaç de crear diferents activitats per a un mateix curs.

Taula 2: Requeriments funcionals

Per tal de que la plataforma funcioni correctament i es pugui adaptar a les necessitats

2.1.2. Requeriments no funcionals

Requeriment	Descripció del requeriment
RNF1	La interfície de la plataforma ha de ser simple, intuïtiva i efectiva.
RNF2	L'ordinador ha de disposar de connexió a internet per tal de connectar-se al servidor.
RNF3	Tant l'ordinador com l'aplicació han de proporcionar temps de resposta ràpids per tal de millorar l'experiència d'usuari.
RNF4	Es necessari disposar de la versió 2016 o posterior de Matlab per a que la plataforma funcioni.

Taula 3: Requeriments no funcionals

2.2. Decisions de disseny

Per tal d'implementar correctament la plataforma s'han agut de prendre una sèrie de decisions de disseny per tal d'adaptar-la a les necessitats dels usuaris (alumnes i professors en aquest cas).

Es dividiran en 2 apartats, les decisions de disseny de la plataforma i el servidor en si, i decisions de disseny dels algoritmes emprats per la plataforma.

2.2.1. Plataforma

Decisions de tecnologies:

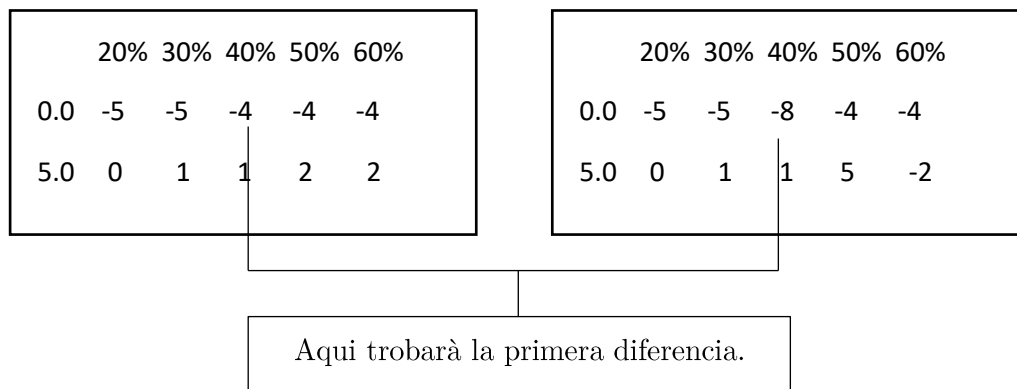
- El servidor que s'utilitzarà serà TomCat Server per tal d'allotjar la plataforma durant el desenvolupament d'aquesta. S'ha decidit utilitzar aquest servidor perquè implementa molt bé la tecnologia JSP (la que s'utilitza en aquesta plataforma).
- La principal tecnologia que s'ha utilitzat per dissenyar la plataforma és JSP. S'ha decidit utilitzar-la perquè, personalment, Java es el meu llenguatge preferit i JSP treballa amb Java. També s'ha decidit utilitzar-la perquè hereta la portabilitat de Java, per tant és possible executar les aplicacions en múltiples plataformes.
- S'ha decidit utilitzar MySQL com a base de dades per a guardar la informació necessària per a que la plataforma pugui funcionar correctament. S'ha decidit utilitzar aquest gestor perquè es un dels més populars per al desenvolupament d'aplicacions web.
- S'haurà d'utilitzar un navegador web per tal d'accedir al servidor. Això es farà mitjançant aquesta adreça web (a on serà necessari accedir amb les credencials de l'alumne o de l'adiminstrador/professor:
http://xxxx:8080/MatLab_ValidatorTool/login.jsp

Decisions funcionals: Totes aquestes decisions s’han pres per tal de poder cobrir correctament tots els requeriments.

- Els arxius pujats pels usuaris es guardaran localment en la maquina que estigui executant la plataforma en aquell moment.
- Aquests arxius es podran trobar en les carpetes mostrades per la pròpia plataforma a l’hora de pujar-los.
- La plataforma generarà un arxiu de “log” a on s’enregistrarà el resultat de la validació. Aquest arxiu es podrà veure a la pàgina de resultats (en format pdf) un cop executat l’algoritme.
- La creació d’usuaris la durà a terme el propi administrador per tal d’evitar problemes interns.
- Només l’administrador/professor podrà crear o borrar activitats.
- La plataforma serà capaç de fer 2 tipus de tests, el test únic i els tests múltiples.
- Per cada tipus de test, la plataforma executarà l’script proporcionat pel professor per tal de poder generar un document de solució adient.
- La plataforma generarà un PDF amb els resultats de la validació.
- El professor podrà escollir si veure o descarregar un PDF resumit o un PDF complet.
- Addicionalment, la plataforma també generarà un fitxer .csv per tal de simplificar la pujada de notes per part del professor.

2.2.2. Algoritmes

- L'algoritme que utilitza la plataforma per fer la validació es una comparació de tuples entre l'arxiu solució i els arxius a validar. Això es gestiona a la funció `getMatch` al `Validator.java`. El que es fa es crear una llista per a cada arxiu amb les tuples d'aquest. Un cop fet això es fa una comparació tupla a tupla a la funció `countMatches` al `Service.java`, que bàsicament mirarà si la paraula de la primera tupla de l'arxiu 1 coincideix amb la paraula de la primera tupla de l'arxiu 2. Es compta el número de tuples que coincideixen i es retorna en forma de percentatge. A continuació es podrà veure un petit exemple amb dos taules reals:



En total, hi ha 3 diferències, per tant el contador tindrà un valor de 15 sobre 18, per tant el percentatge de semblança serà d'un 83,3%.

La sortida de l'algoritme és la següent (suposant que el primer arxiu és diu `T1_sol` i el segon `Psol`):

Comparing T1_sol with Psol

Percentage of match: 68.0%

S'ha decidit fer-ho així perquè es una de les millors formes (sobretot en eficiència) de comparar dos arxius.

- El resultat de la validació s'imprimirà en percentatge. Aquest valor farà referència al percentatge de semblança que hi ha entre els dos arxius (el de solució i el que es vol validar, tal i com s'ha explicat en la decisió funcional anterior). Un exemple de la sortida d'una validació múltiple seria la següent:

Comparing T2_sol.txt with 0.txt

Percentage of match: 9.0%

Comparing T2_sol.txt with 1.txt

Percentage of match: 69.0%

Comparing T2_sol.txt with 2.txt

Percentage of match: 69.0%

- S'han utilitzat funcions i classes bàsiques de java per tal d'accedir a la base de dades (Connection i PreparedStatement).. Per tal de que java sigui capaç de connectar-se amb una BD, s'utilitza la classe connection, que passant-li la url de la BD, el seu username i la seva password, es podrà crear la connexió. Un cop creada es poden generar sentències SQL amb la classe PreparedStatement.

- Per tal de poder executar Matlab des de Java s'utilitza la classe Process, que l'executarà mitjançant la línia de comandes. El que fa es crear un procés amb la funció getRuntime de la classe Runtime, tal i com es pot veure en aquesta crida:

```
Process p = Runtime.getRuntime().exec(command);
```

A on command es la comanda que s'executarà, que en aquest cas podria ser:

```
/Applications/MATLAB_R2017a.app/bin/matlab -nodesktop  
-nosplash -r  
run('/Users/alex/Documents/TFG/Tests/Tests/2/test_plab11  
617_tarda_test2'); -logfile  
/Users/alex/Documents/TFG/Tests/wmlog.txt"
```

A on s'indica que es vol executar l'arxiu test_plab11617_tarda_test2 i es vol generar un logfile a l'adreça indicada, amb les opcions -nodesktop i -nosplash per tal de que no s'obri cap instància de Matlab visible per a l'usuari.

El principal avantatge de fer-ho d'aquesta manera es que es bastant fàcil d'implementar. El principal inconvenient es que tarda un temps bastant important en fer la execució, ja que Matlab no està massa ben optimitzat per ser cridat a través de la línia de comandes per fer la execució.

Hi ha altres formes de cridar Matlab des de Java (per exemple: la llibreria MatlabEngine), però, per problemes de compatibilitat s'ha decidit fer-ho d'aquesta manera.

2.3. Disseny de la base de dades

Per tal de que la plataforma sigui capaç d'enregistrar la informació requerida, es va determinar que s'havia d'emprar una base de dades. Com s'ha explicat en les decisions de disseny, s'ha decidit utilitzar MySQL. Encara així, la plataforma no necessita de grans taules de BD per emmagatzemar la seva informació, ja que només empra la BD per gestionar els logins i les activitats.

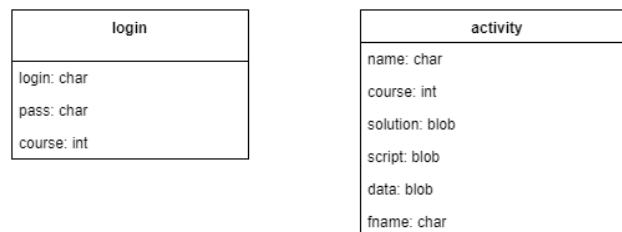


Figure 1: Diagrama BD

En aquest diagrama es pot veure l'organització de les 2 taules la plataforma utilitza per tal de guardar correctament totes les dades. S'ha de mencionar que aquestes 2 taules NO tenen cap relació entre si, son totalment independents entre elles. A continuació s'explicaran amb més detall les taules implementades en la DB.

2.3.1. Taula de login

Aquesta taula s'encarrega d'emmagatzemar la informació sobre els usuaris de la plataforma (tant alumnes com professors). Té els següents camps:

- **login:** Atribut de tipus char. És el nom que l'usuari ha escollit per tal de poder accedir a la plataforma.

- **pass:** Atribut de tipus char. És la contrasenya associada al nom de l'usuari.
- **course:** Atribut de tipus int que guardarà el curs que està cursant l'alumne actualment.

2.3.2. Taula de activity

Aquesta taula serà de vital importància per tal de poder tractar correctament les activitats. Els seus atributs son els següents:

- **name:** Atribut de tipus char. És el nom que se li assigna a cada activitat. Aquest nom s'utilitzarà també en el procés d'eliminació de les activitats.
- **course:** Atribut de tipus int. És el curs (que podrà ser, 1, 2, 3 o 4) en el que l'activitat està associada (en l'apartat de rols 2.4 això s'explicarà millor).
- **solution:** Atribut de tipus blob. És la solució que s'ha assignat a l'activitat. S'ha decidit que sigui de tipus blob per a poder pujar/descarregar fàcilment el fitxer utilitzant java.
- **script:** Atribut de tipus blob. Guardarà l'script que el professor ha d'haver proporcionat per tal de que la plataforma l'executi i pugui generar el document de sortida.
- **data:** Atribut de tipus blob. Guardarà les dades necessàries per tal de que l'script pugui executar-se correctament i ho faci amb certes condicions establertes prèviament pel professor.

- **fname:** Atribut de tipus char. Guardarà el nom de l'arxiu de dades per a que, a l'hora de fer l'import en l'script, no doni cap error de nomenclatura.

2.4. Rols

La plataforma serà capaç de gestionar 2 rols principals:

- **Rol del professor/administrador:** Aquests tindran unes credencials especials per tal d'accedir a la plataforma. La seva pantalla d'inici (explicades amb més profunditat en l'apartat 3 d'aquest document) consistirà en els 4 possibles cursos, a on, seleccionant-ne un, s'accedirà a la pantalla d'activitats d'aquest.
L'administrador/professor serà capaç de crear i eliminar activitats i de fer validacions múltiples (Multiple activity tests). La seva pantalla de resultats disposarà de 3 possibles documents: el simplificat, el complet o el .csv (utilitzat per ficar notes).
- **Rol de l'estudiant:** Aquests estaran assignats a un curs. Una vegada inicien sessió, els hi apareixerà la pantalla de les activitats del seu curs assignat. Els alumnes podran fer validacions úniques (Single activity tests). En cap cas podran eliminar o crear activitats.

2.5. Diagrama de classes

A continuació es podrà veure el diagrama de classes de la plataforma. Està dividit principalment en 2 seccions, els .java (servlets) i els .jsp.

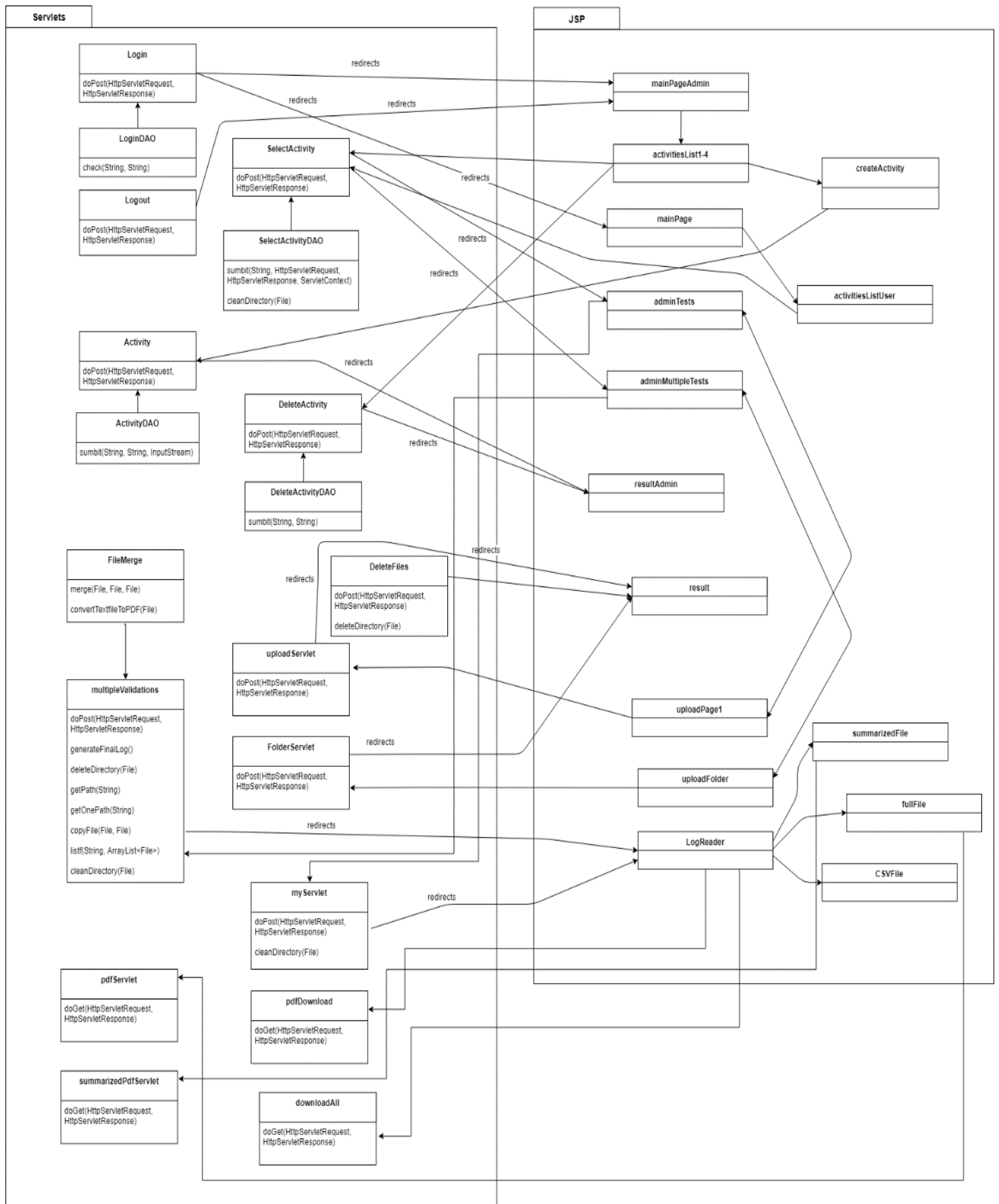


Figure 2: Diagrama de classes de la plataforma

3. Disseny de la interfície

La interfície no és un dels objectius principals del projecte ja que s'ha prioritzat el funcionament de la plataforma. Per falta de temps no s'ha pogut abordar la creació d'una interfície amb tecnologies visuals habituals, com poden ser: jQuery UI, Webix, jqWidgets...

S'ha intentat dissenyar una interfície eficaç, ràpida i intuïtiva (sense entrar en detall amb les tecnologies mencionades anteriorment). Per fer-ho, s'analitzarà cada pantalla de la plataforma en el següents sub-apartats.

3.1. Interfícies del professor/administrador

La interfície del login es bastant senzilla, només requereix del username i de la password assignada al username.

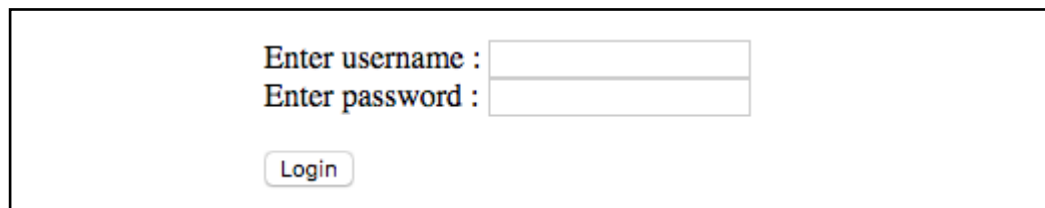
A screenshot of a login form. It features two input fields: the first is labeled 'Enter username :' and the second is labeled 'Enter password :'. Below these fields is a button labeled 'Login'. The entire form is enclosed in a thin black rectangular border.

Figure 3: Interfície de login

A continuació es podrà observar la pantalla principal un cop iniciada la sessió:

Matlab Validator Tool

Hi, admin!

Here you'll be able to select the course you want to handle:

1	2	3	4
Select	Select	Select	Select

Logout

Figure 4: Interfície principal del professor/administrador

En aquesta pantalla el professor podrà escollir un dels 4 cursos per poder gestionar les activitats. Un cop seleccionat el curs el professor accedirà a la pantalla que es pot veure en la Figura 5.

Activities List

Create an activity:

Create

Activity name	Course	Delete Activity	Single activity test	Multiple activity tests
Practica JFLAP	2	Delete Activity	Test Activity	Test Activity
Practica Yacc	2	Delete Activity	Test Activity	Test Activity

Go back

Figure 5: Interfície de la llista d'activitats

La pàgina d'activitats mostrarà en una taula totes les activitats del curs seleccionat. Principalment hi hauran 3 opcions:

- **Delete Activity:** Borarà l'activitat seleccionada.
- **Single activity test:** Et portarà a una pàgina per gestionar tests d'un en un.
- **Multiple activity tests:** Et portarà a una pàgina per gestionar múltiples tests a la vegada.

Adicionalment també hi ha un botó per crear activitats amb la següent interfície:



The image shows a web form titled "Create activity" in a large, bold, black serif font. Below the title, there are several input fields and buttons. The first two are text inputs: "Enter the name of the activity :" followed by a long rectangular box, and "Enter its course :" followed by a shorter rectangular box. Below these are three upload sections, each consisting of a label, a button, and a status text. The first is "Upload solution :" with a button labeled "Seleccionar archivo" and the text "Ningún archivo seleccionado". The second is "Upload script :" with a button labeled "Seleccionar archivo" and the text "Ningún archivo seleccionado". The third is "Upload data :" with a button labeled "Seleccionar archivo" and the text "Ningún archivo seleccionado". Below these is a text input field labeled "Write name of the file data :". At the bottom of the form are two buttons: "Submit" and "Go back", both with rounded corners and a light gray background.

Figure 6: Interfície de creació d'activitats

A on, obligatòriament s'haurà d'introduir el nom de l'activitat i el curs al qual pertany. Per tal de realitzar les proves correctament s'hauria de proporcionar la solució final, el script d'execució i, opcionalment, les dades addicionals que el programa ha d'emprar i el nom del fitxer de dades per motius interns de la plataforma.

A continuació es podrà veure la pantalla de la Single activity test, pensada per poder gestionar una sola validació:

Single activity test page

Instruction	Button
Step 1: Sumbit the solution you want to test	<div>Upload code</div>
Step 2: Press Validate	<div>Validate</div>

Note: All the uploaded files will be deleted after pressing the "Validate" button.

Go back

Figure 7: interfície de Single activity test

Aquí l'usuari haurà de pujar el codi que voldrà testejar en l'step 1, i per tal de validar-lo haurà d'apretar "Validate" en l'step 2.

A continuació es podrà veure la pantalla des Multiple activity tests, pensada per poder gestionar múltiples validacions a l'hora:

Multiple activity tests page

Instruction	Button
Step 1: Sumbit the solution you want to test	<div>Upload code</div>
Step 2: Press Validate	<div>Validate</div>

Note: All the uploaded files will be deleted after pressing the "Validate" button.

Go back

Figure 8: Interfície de Multiple activity test

Aquí l'usuari haurà de pujar el codi apretant el botó “Upload code” de l'step 1, que el redirigirà a una pàgina a on, introduint la direcció de les carpetes (per exemple: Users/Admin/Documents/TFG/Tests), es pujarà el codi a la plataforma.

Un cop començada la validació, l'usuari haurà d'esperar a que aquesta finalitzi. Un cop la validació ha finalitzat apareixerà la següent pàgina de resultats:

Results page

Type of file	See the file	Download file
Summarized file	See file	Download file
Full file	See file	Download file
CSV file	See file	Download file

Note: The CSV file is only generated for the multiple tests

[Download all files](#)

[Go back](#)

Figure 9: Interfície de resultats

En aquesta pàgina es podran fer diverses accions (descarregar o veure l'arxiu) sobre 3 arxius generats:

- **Summarized file:** Es un arxiu resumit que mostra el percentatge de semblança amb l'arxiu de solució del professor.
Exemple:
Percentage of match: 72.0%

- **Full file:** Aquest arxiu es el “complet”. Per a cada alumne validat es mostra el percentatge de semblança i l’output que ha generat Matlab un cop executat el seu codi amb l’script del professor. Una cosa a destacar es que quan es fan validacions múltiples, al començament de cada resultat apareix el nom de l’estudiant, cosa que en el següent exemple no apareixerà perquè forma part d’una validació única.

Exemple:

Percentage of match: 72.0%

< M A T L A B (R) >

Copyright 1984-2017 The MathWorks, Inc.

R2017a (9.2.0.538062) 64-bit (maci64)

February 23, 2017

To get started, type one of these: helpwin, helpdesk, or demo.

For product information, visit www.mathworks.com.

```
0.0C -5 -5 -4 -4 -4 -3 -3 -2 -2
5.0C 0 1 1 2 2 3 3 4 4
10.0C 6 6 7 8 8 9 9 10 11
15.0C 11 12 13 14 14 15 16 17 18
20.0C 17 18 19 20 21 22 23 24 25
25.0C 22 24 25 27 28 30 31 32 34
30.0C 28 30 32 34 36 38 39 41 43
35.0C 34 37 39 41 44 46 49 51 54
40.0C 41 44 47 50 53 56 59 62 65
45.0C 47 51 55 59 63 67 71 74 78
50.0C 54 59 64 69 74 79 84 88 93
55.0C 62 68 74 80 86 92 98 104 111
60.0C 70 77 85 92 100 107 115 123 130
65.0C 78 87 97 106 115 125 134 143 153
70.0C 87 99 110 122 133 145 156 167 179
75.0C 97 111 125 139 153 167 181 195 209
80.0C 108 125 142 159 176 193 209 226 243
85.0C 120 140 161 181 201 222 242 262 283
90.0C 133 158 182 206 231 255 279 304 328
95.0C 148 177 206 235 264 293 322 351 380
```

perillExtrem =

110

perill =
6
precaucio =
13
confort =
51

----- RESUM GRAUS CONFORT -----

4 - Perill extrem : 110 [61.11%]
3 - Perill : 6 [3.33%]
2 - Precauci : 13 [7.22%]
1 - Lmit de confort : 51 [28.33%]

NOMBRE VALORS : 180 [100.00%]

- **CSV file:** Arxiu que servirà al professor per posar les notes finals a cada alumne. Té el següent format:

arm13,"Rovira Martí Albert", 78.0

A on "arm13" es el codi de l'alumne, "Albert Rovira Martí" es el seu nom, i 78.0 el percentatge de semblança amb la solució del professor.

Exemple:

```
Student ID,Student Name,Percentage of Match  
rsr3,"Sumalla Rosell Roger",24.0  
ttv1,"Tello Vidal Tots",72.0  
jtb5,"Tuneu Busquets Jordi",64.0
```

Nota important: s'ha de tenir en compte que totes aquestes pantalles formen part de la interfície del professor. La interfície de l'alumne es molt més senzilla. Tenint en compte que un alumne està assignat a un curs, l'alumne, a l'iniciar sessió només veurà les activitats del seu curs, i només podrà fer un single test (òbviament no podrà crear ni borrar activitats).

3.2. Diferències en les interfícies de l'alumne

L'alumne, per norma general, no haurà de ser capaç de fer certes accions que el professor pot fer. Per això, les interfícies de l'alumne son bastant més limitades. Les principals diferències són les següents:

- La seva pantalla principal serà la de les activitats del curs que l'alumne té assignat (ja que s'ha decidit que un alumne no pot estar assignat a més d'un curs a l'hora).
- Per motius obvis, l'alumne no serà capaç de crear o eliminar activitats a la pàgina d'activitats.
- L'alumne només podrà realitzar Single activity tests (no té sentit que pugui fer-ne molts a l'hora, ja que només fa 1 cop l'exercici).

4. Implementació

En aquest apartat es parlarà sobre els components més importants de la plataforma, així com el software emprat per al disseny i el desenvolupament d'aquesta. En cada un dels components s'explicaran els motius que s'han tingut en compte per tal d'escollir-lo.

4.1. Llenguatge emprat: Java

Java és un llenguatge de programació dissenyat el 1990 per James Gosling amb altres companys de Sun Microsystems a partir del llenguatge C. Des del seu naixement fou pensat com un llenguatge orientat a objectes. Entre el 13 de novembre de 2006 i el maig del 2007 Sun va alliberar parts de Java com a programari lliure de codi obert amb llicència GPL. És un dels llenguatges de programació més utilitzats, i s'utilitza tant per aplicacions web com per aplicacions d'escriptori.



S'ha decidit escollir Java com a llenguatge principal de la plataforma perquè, al ser un llenguatge interpretat, ofereix un índex de re-utilització de codi molt elevat, sent possible trobar moltes llibreries lliures de Java. És un llenguatge flexible i potent tot i la facilitat amb la qual es programa i dels resultats que ofereix. Un dels trets que el caracteritza i que el fa una eina molt valorada a l'hora de desenvolupar aplicacions distribuïdes, és el fet que és un llenguatge multi-plataforma.

4.2. JSP

JavaServer Pages (JSP) és una tecnologia que permet als desenvolupadors de pàgines web, generar respostes dinàmicament a peticions HTTP. La tecnologia permet que codi Java i certes accions predefinides siguin incrustades en un context estàtic. Els JSPs són compilats en Servlets per un compilador JSP. Aquest pot generar un servlet o generar bytecode directament.

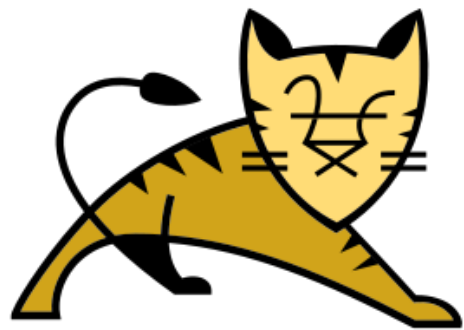


Un altre aspecte important de JSP es que disposa dels seus famosos tags XML addicionals, anomenats accions de JSP, s'utilitzen per invocar altres funcions. Addicionalment, la tecnologia permet la creació de llibreries d'etiquetes que actuen com extensions de l'estàndard d'etiquetes HTML o XML. Les llibreries d'etiquetes aporten una forma multiplataforma d'ampliar les capacitats d'un servidor web.

S'ha escollit aquesta tecnologia perquè hereta la portabilitat de Java, per tant és possible executar les aplicacions en múltiples plataformes.

4.3. Apache TomCat Server

Apache Tomcat és un contenidor de servlets desenvolupat a l'Apache Software Foundation. Tomcat implementa les especificacions de servlet i de JavaServer Pages (JSP) de Sun Microsystems, proporcionant un entorn per al codi Java a executar en cooperació amb un servidor web.



Aquest afegeix eines per a la configuració i el manteniment, però també pot ser configurat editant els fitxers de configuració que normalment són en format XML. Tomcat inclou el seu propi servidor HTTP, per això també se'l considera un servidor webindependent.

S'ha decidit utilitzar aquesta tecnologia perquè inclou un compilador Jasper, que compila JSPs convertint-los en servlets.

4.4. Entorn de desenvolupament

Per tal d'implementar la plataforma s'ha decidit utilitzar un IDE (integrated development environment) anomenat NetBeans, ja que és de codi obert, lliure, gratuït i, principalment, està fet per al llenguatge de programació Java.

Un altre punt a favor de Netbeans es que té una gran base d'usuaris, amb una comunitat de constant creixement amb prop de 100 socis a tot el mon, cosa que afavoreix el poder buscar una solució als diferents problemes que sorgeixin a l'hora d'implementar un projecte.



5. Conclusions

Per tal de finalitzar aquest treball, a continuació es tractaran tant les conclusions obtingudes com els possibles treballs futurs o possibles ampliacions que es podrien fer.

5.1. Conclusions finals

S'ha fet una plataforma que permet, per un costat, als alumnes provar les seves solucions i rebre un feedback immediat en forma de document, que donarà un percentatge de semblança amb la solució del professor, permetent-li així autoavaluar-se. Per altra banda, permet al professor crear una sèrie d'activitats i tenir una nota en funció de la qualitat de la solució de l'alumne.

Una de les principals funcionalitats de la plataforma és que permet fer múltiples validacions a l'hora per tal d'automatitzar la correcció d'exercicis i poder estalviar-li molt de temps al professor.

Tal i com s'ha especificat anteriorment, es va cometre un error irreparable durant la implementació de l'aplicació, limitant la plataforma a 4 cursos (pensat per a un grau de 4 anys). Per tal de millorar aquest aspecte, el professor hauria d'haver disposat de la capacitat de crear o eliminar cursos.

Gràcies al desenvolupament d'aquest projecte he aconseguit millorar el meu nivell amb el llenguatge de programació Java, aprendre a utilitzar els JSPs (Java Server Pages) i aprendre a utilitzar una base de dades MySQL, tot això amb l'entorn de treball NetBeans.

Una de les principals conclusions que s'ha obtingut gràcies a aquest treball és que el temps és un dels factors més importants a l'hora de dissenyar i implementar qualsevol aplicació o projecte software.

Una altra conclusió que s'ha pogut extreure es que les fases més importants d'un projecte són la d'anàlisis i disseny. Això es perquè un bon anàlisis i disseny fan que es puguin evitar una enorme quantitat d'errors durant la implementació de la aplicació.

Respecte a la memòria del projecte, no m'ha suposat cap dificultat redactar-lo, ja que considero que escriure un informe sobre el desenvolupament d'una aplicació és un dels meus punts forts (obviant el fet de que gaudeixo fent-los), com ho hauria de ser per a qualsevol enginyer informàtic, a causa de la necessitat que hi ha de documentar correctament el que es desenvolupa. He intentat redactar un document que permeti a gairebé qualsevol lector (amb una mínima base informàtica) entendre com s'ha implementat el projecte.

5.2. Valoració i autocrítica

A continuació es farà una valoració del seguiment que es va proposar a l'inici del treball:

- **Preparació del projecte:** Les tasques d'aquest apartat es van fer al principi (gener i febrer). Es van poder completar a temps, ja que eren tasques bastant senzilles (instal·lar recursos, estudiar-los, planificar el projecte...).
- **Desenvolupament del projecte:** Aquestes altres tasques van requerir de molta més dedicació (des de el febrer fins al juny). Implementar tota la plataforma em va comportar més hores de les que havia determinat, juntament amb els tests i el desplegament d'aquesta.
- **Presentació del projecte:** Un bon desenvolupament sempre ha d'estar ben documentat i presentat, per tant vaig tardar aproximadament un mes en redactar correctament la memòria i poder documentar el desenvolupament.

També considero que es necessari fer una mica d'autocrítica i crec que a aquest projecte li falten dos coses essencials per ser millorar-lo significativament:

- Una interfície molt més atractiva: La que té ara és molt plana i senzilla principalment perquè no era prioritària ja que el temps per dissenyar una bona interfície no s'ajusta al temps d'aquest projecte i per tant no ho he abordat.
- Al principi del treball no es va tenir en compte la possibilitat de modificar la quantitat de cursos, i es va determinar que n'hi haurien 4, amb les seves respectives activitats. La forma correcta de fer-ho era poder-ne crear per tal de que la plataforma es mantingués actualitzada amb el pas del temps.

5.3. Treballs futurs o possibles ampliacions

Al finalitzar la realització de projecte, seria adient analitzar les possibles ampliacions d'aquest:

- Millorar l'apartat gràfic de la plataforma. Que disposi d'una interfície molt més dinàmica i atractiva.
- Corregir l'organització de cursos, ja que s'hauria de poder-ne crear amb facilitat. Que cada alumne sigui capaç d'inscriure's a un curs i poder-ne veure les activitats.
- Que la plataforma fos capaç d'analitzar el codi enviat pels alumnes i determinar si segueixen una sèrie de normes d'estil de Matlab, avisant-los en cas contrari.
- A l'hora de proposar el tema del treball, és va plantejar primerament un possible detector de plagis.
La plataforma en si es capaç de validar les solucions dels alumnes, però una molt bona ampliació seria que fos capaç de determinar si existeix plagi entre aquestes entregues.

6. Bibliografia

- [1] Java. Oracle Docs. <https://docs.oracle.com/javase/tutorial/>

- [2] NetBeans. Developing Web Applications.
<https://netbeans.org/kb/docs/web/quickstart-webapps.html>

- [3] Wikipedia. JSP. https://es.wikipedia.org/wiki/JavaServer_Pages

- [4] TutorialsPoint. Implementing JSP on NetBeans IDE.
<https://www.tutorialspoint.com/articles/implementing-jsp-on-netbeans-ide>

- [5] NetBeans. Connecting to a MySQL Database.
<https://netbeans.org/kb/docs/ide/mysql.html>

- [6] GitHub. https://github.com/malerax213/Matlab_ValidatorTool_JSP

- [7] CodeJava. Upload files to Database.
<http://www.codejava.net/coding/upload-files-to-database-servlet-jsp-mysql>

- [8] CodeJava. Read File data from Database. <http://www.codejava.net/java-se/jdbc/read-file-data-from-database-using-jdbc>